REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE Technical Report	3. DATES COVERED (From - To) -

4. TITLE AND SUBTITLE ADEN: Anomaly Detection Engine for Networks	5a. CONTRACT NUMBER
	5b. GRANT NUMBER W911NF-11-C-0215
	5c. PROGRAM ELEMENT NUMBER 1M30BM
6. AUTHORS V.S. Subrahmanian	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Maryland - College Park Research Administration 3112 Lee Building College Park, MD 20742 -5141	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) 60132-NS-DRP.1

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

The ADEN team completed its first project year with a successful evaluation of the preliminary version of its detection engine. The advancing availability of data had a major influence on the direction of our work. We started with public data from Wikipedia for adversary detection by content analysis. With the availability of the synthetic datasets generated by CERT, we refocused our work to address relational data. Finally, the more comprehensive SureView collected at Raytheon gives us the opportunity to extend our anomaly detection engine with the design of

15. SUBJECT TERMS

Anomaly Detection Engine

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON V.S. Subrahmanian
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 301-405-6724

Report Title

ADEN: Anomaly Detection Engine for Networks

ABSTRACT

The ADEN team completed its first project year with a successful evaluation of the preliminary version of its detection engine. The advancing availability of data had a major influence on the direction of our work. We started with public data from Wikipedia for adversary detection by content analysis. With the availability of the synthetic datasets generated by CERT, we refocused our work to address relational data. Finally, the more comprehensive SureView collected at Raytheon gives us the opportunity to extend our anomaly detection engine with the design of a Combined Codebook consisting of a mix of textual, relational, and network oriented variables that may all be linked to identifying insider threat.

ADEN: ANOMALY DETECTION ENGINE FOR NETWORKS

ANNUAL REPORT – 1st YEAR

V.S. Subrahmanian
Dept. of Computer Science & UMIACS
University of Maryland
College Park, MD 20742
vs@cs.umd.edu

301-405-6724

ABSTRACT

The ADEN team completed its first project year with a successful evaluation of the preliminary version of its detection engine. The advancing availability of data had a major influence on the direction of our work. We started with public data from Wikipedia for adversary detection by content analysis. With the availability of the synthetic datasets generated by CERT, we refocused our work to address relational data. Finally, the more comprehensive SureView collected at Raytheon gives us the opportunity to extend our anomaly detection engine with the design of a Combined Codebook consisting of a mix of textual, relational, and network oriented variables that may all be linked to identifying insider threat.

I. TECHNICAL DETAILS

The goal of the ADEN project is to identify individuals within a trusted network who misbehave – either by leaking classified data or otherwise. In our initial proposal, we had planned to work with Wikipedia data in order to identify Wikipedia “vandals” (individuals who falsify Wikipedia entries). As the ADAMS project proceeded, we were asked to work on different data sets including relational data sets from CERT and from Raytheon.

As the available data had a big influence on our algorithms we organize this report along the datasets.

OPEN DATA: WIKIPEDIA

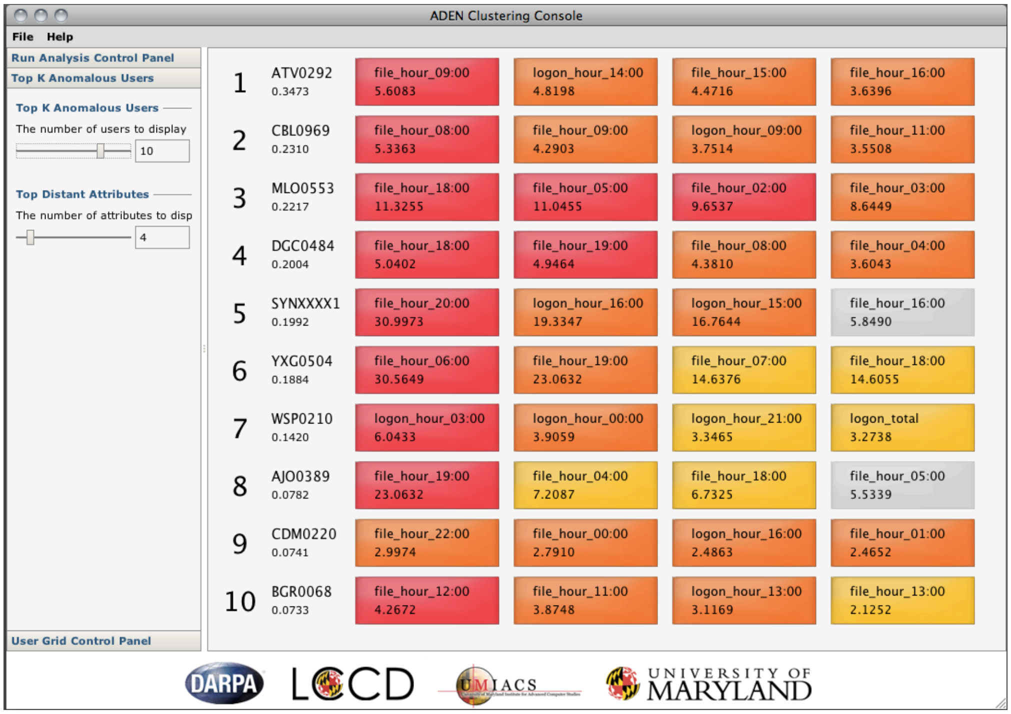
Our initial intention was to build an anomaly detection engine based on the analysis of the document a user views. Shifts in the topics a user shows interest for and interest in documents that other users with the same mission do not view indicate suspicious behavior. As a replacement for data on system usage we created a Wikipedia vandalism data set and started with vandalism detection on Wikipedia.

We developed a method by which we can use histograms to characterize the normal behavior of a user. For instance, suppose we have a training set TS (e.g. last 2 months of documents accessed by user U). Moreover, suppose T1,..,Tn are the topics (named entities) that occur in TS. For each topic Ti, we can characterize the significance Si of the occurrence of Ti in TS through standard text analysis (e.g. frequency analysis, TF-IDF measures, etc). The behavior of a user u is now defined as a histogram H(u) over these topics, explaining the relative importance of each topic Ti in the set of documents accessed by the user.

We used the distance between a user and a “normal” user as the measure for anomality. This algorithm showed good results on several test datasets.

CERT DATA

The initial synthetic datasets provided by CERT are log files of user activity. We developed a codebook of variables that describe the log file data. The codebook contains for example variables for the average number of log on events by hour, the average number of files copied to thumb-drives by hour. In our initial approach (see Figure 1) we generated for every variable averages over the complete time horizon of the dataset and experimented with clustering approaches to identify outliers. Each user was represented by a vector of codebook values. Users that are not in dense regions of the multi-dimensional space spanned by the user vectors have been regarded as outliers. This approach did not satisfactory precision. Despite efforts to handle the dimensionality problem (we experimented with dimensionality reduction) we did not get satisfactory results. The low noise/information ratio showed to be a major problem. Furthermore, users showed to have inhomogeneous behaviors. Non-standard behavior with respect to the total user base seems to be no good indicator for malicious behavior.



The number of users to display 10

The number of attributes to disp 4

1	ATV0292 0.3473	file_hour_09:00 5.6083	logon_hour_14:00 4.8198	file_hour_15:00 4.4716	file_hour_16:00 3.6396
2	CBL0969 0.2310	file_hour_08:00 5.3363	file_hour_09:00 4.2903	logon_hour_09:00 3.7514	file_hour_11:00 3.5508
3	MLO0553 0.2217	file_hour_18:00 11.3255	file_hour_05:00 11.0455	file_hour_02:00 9.6537	file_hour_03:00 8.6449
4	DGC0484 0.2004	file_hour_18:00 5.0402	file_hour_19:00 4.9464	file_hour_08:00 4.3810	file_hour_04:00 3.6043
5	SYNXXXX1 0.1992	file_hour_20:00 30.9973	logon_hour_16:00 19.3347	logon_hour_15:00 16.7644	file_hour_16:00 5.8490
6	YXG0504 0.1884	file_hour_06:00 30.5649	file_hour_19:00 23.0632	file_hour_07:00 14.6376	file_hour_18:00 14.6055
7	WSP0210 0.1420	logon_hour_03:00 6.0433	logon_hour_00:00 3.9059	logon_hour_21:00 3.3465	logon_total 3.2738
8	AJO0389 0.0782	file_hour_19:00 23.0632	file_hour_04:00 7.2087	file_hour_18:00 6.7325	file_hour_05:00 5.5339
9	CDM0220 0.0741	file_hour_22:00 2.9974	file_hour_00:00 2.7910	logon_hour_16:00 2.4863	file_hour_01:00 2.4652
10	BGR0068 0.0733	file_hour_12:00 4.2672	file_hour_11:00 3.8748	logon_hour_13:00 3.1169	file_hour_13:00 2.1252

Figure 1: ADEN Clustering Console

From what we learned from the global outlier analysis, we regard completely unsupervised approaches to identifying attackers and attacks to be less promising. For example, learning the predictive power of different types of user behavior is hardly possible. Therefore, we investigated other approaches that require some user input. User inputs are a way to integrate external information.

Based on the experience of the first prototype, we designed a new detector engine. Our key design decisions were:

- Measure anomaly by temporal changes in the behavior of a user
- Address dimensionality problem by directly selecting promising dimensions
- Put more emphasis on user-system interaction: because of the expected high false positive rate of the detection system, the user needs to get actionable information why a user raised suspicion

For our new approach we continue to extract the codebook variables (1st degree variables) from the raw log data (CERT or SureView). Then we derive from these variables so-called 2nd degree variables. These 2nd degree variables encode indicators of insider attacks as well as indications that a user has an increased proneness to conduct attacks.

We reviewed literature on insider attacks to identify what makes a user become an attacker. From the literature review we created a list of symptoms (e.g. professional or private stress) and several indicators for each symptom. In this way we will complement the detection of abnormal behavior with information on not uncommon but worrisome behavior. We think we will make best use of the sparse information in the large pool of data by this combined analysis of symptoms and abnormal behavior.

From the literature review we derived four major indicators that are correlated with the probability that a user starts an insider attack:

- Start of Employment
- Termination of Employment
- Personal Problems (illness/death in family, marital/relationship problems)
- Professional Problems (negative changes at workplace, interpersonal conflicts)

Furthermore, we encode in 2nd degree variables indicators that are directly related to attacks, for example:

- Increased File Access
- Copy of Executable Files to a Computer
- Unusual working hours

Our system raises alerts, when more than a user-defined number of 2nd degree variables indicate an attack. This way, users can configure the sensitivity of the system.

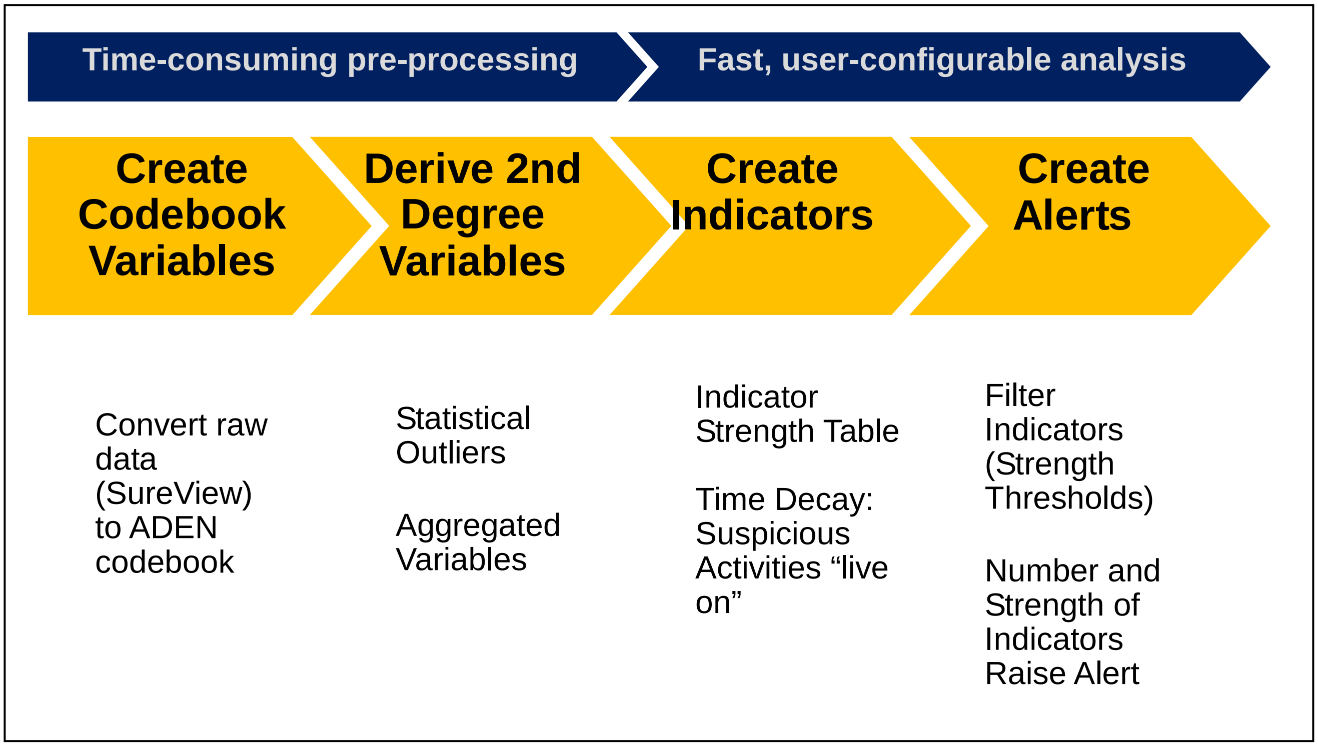

Time-consuming pre-processing
Fast, user-configurable analysis
Create Codebook Variables
Derive 2nd Degree Variables
Create Indicators
Create Alerts
Convert raw data (SureView) to ADEN codebook
Statistical Outliers
Aggregated Variables
Indicator Strength Table
Time Decay: Suspicious Activities “live on”
Filter Indicators (Strength Thresholds)
Number and Strength of Indicators Raise Alert

Figure 2: Process Flow ADEN Detector Engine

Dataset	Alerts	Suspicious Users	Detected Incidents
Cert3v1	44	29	2 / 2
Cert3v2	46	33	2 / 2
Cert4v1	46	25	2 / 3
Cert4v2	139	75	26 / 30

Table 1: Evaluation Results ADEN Detector Engine

Figure 2 shows the analysis process of the ADEN Detector Engine. In the two first steps codebook variables are generated and used to create 2nd degree variable. The next two steps are fast and user-configurable to give the user interactive feedback. This process flow bundles the computational expensive tasks in pre-processing steps, so that the system can provide quick responses to user inputs.

The results of this approach are very promising (see Table 1). We analyzed 4 datasets and identified with standard parameter settings most incidents with a good precision (for this kind of problem).

SUREVIEW DATA

From the beginning, the synthetic CERT data has been generated with the objective to resemble the SureView data and with every new release the CERT data included more features of the SureView data. Our work on the SureView data is a direct continuation of the work with the CERT data.

As soon as we started work with the CERT data, we realized that in order to have a reasonable chance of successfully finding insider threat, we need to track numerous variables:

- Textual variables – as in our original Wikipedia based work;
- Relational variables – as in the CERT data work;
- Network variables – as in our work related to our original proposal which were not explicitly incorporated into either our CERT or our Wikipedia based work.

In the Wikipedia-based work, we only deal with textual variables, while the CERT work, we only dealt with relational variables. We in the process of defining a *Combined Codebook* that captures a set of variables incorporating all three types of variables when addressing the SureView data.

As textual and relational variables have already been discussed, we briefly discuss the types of network variables we hope to incorporate into ADEN.

- *Variables relating to centrality measures*. There are many ways in which centrality measures can be defined w.r.t. a behavior network B**N** consisting of users, documents, and topics, and edges consisting of user-user links (which users know which other users), user-document links (which users read a given document) and topic-document links (which topics are present in a document). For instance, we can define a notion of between-ness centrality of a “topic” node *t* w.r..t a user *u* by trying to understand how central *t* is w.r.t. the restriction of network **N** to a subgraph associated with user *u* and the documents/topics he has accessed. This can be done in many ways. We are leveraging work in our lab on computation of between-ness centrality and work on eigenvector centrality for this purpose. The hypothesis is that if the centrality of topics that a user is interested in change a lot over time, or if they vary a lot (high standard deviation), then the user may be more suspicious. Alternatively, certain topics (e.g. baseball playoffs) may be central to a very large number of users, allowing us to eliminate these topics as “noisy” topics. In addition to between-ness and eigenvector centrality, we are interested in adapting the newly introduced notion of “covertness centrality” of nodes in **BN** to identify potential covert activity.
- *Variables related to certain subgraph patterns and importance scores*. Analysts and law enforcement officials have a great deal of expertise on suspicious behavioral patterns. We want to be able to leverage this information in our search for insider threat. Such patterns can often be expressed as subgraph queries to the behavior network **BN**. We are leveraging work in our lab on ranking patterns in graph data

(such as **BN**) and using the importance scores of found patterns (as sound discovered patterns may be a better match for a query subgraph than other found patterns) to define a set of variables to add to our codebook.

- *Variables related to gaming the adversary*. We also want to look at variables where we place ourselves in the shoes of the bad guy (i.e. the crooked insider). Here, we want to identify specific topics or documents that a bad guy may go after which would allow him to minimize the prospect of discovery while still achieving a goal of getting information he, the user, is not authorized to get. This falls within a class of problems called "social network optimization problems" in which we want to find *k* nodes in a network having some property or properties (e.g. being topic or document nodes and also being outside the user's mission) that optimize an objective function (e.g. minimize the bad guy's probability of being discovered). Thus, by putting ourselves in the shoes of the adversary, we hope to be able to detect sets of *k* nodes such that if a user *u* were to access these nodes, our suspicion of him would increase. Leveraging these results, we are in the process of defining a new set of variables.

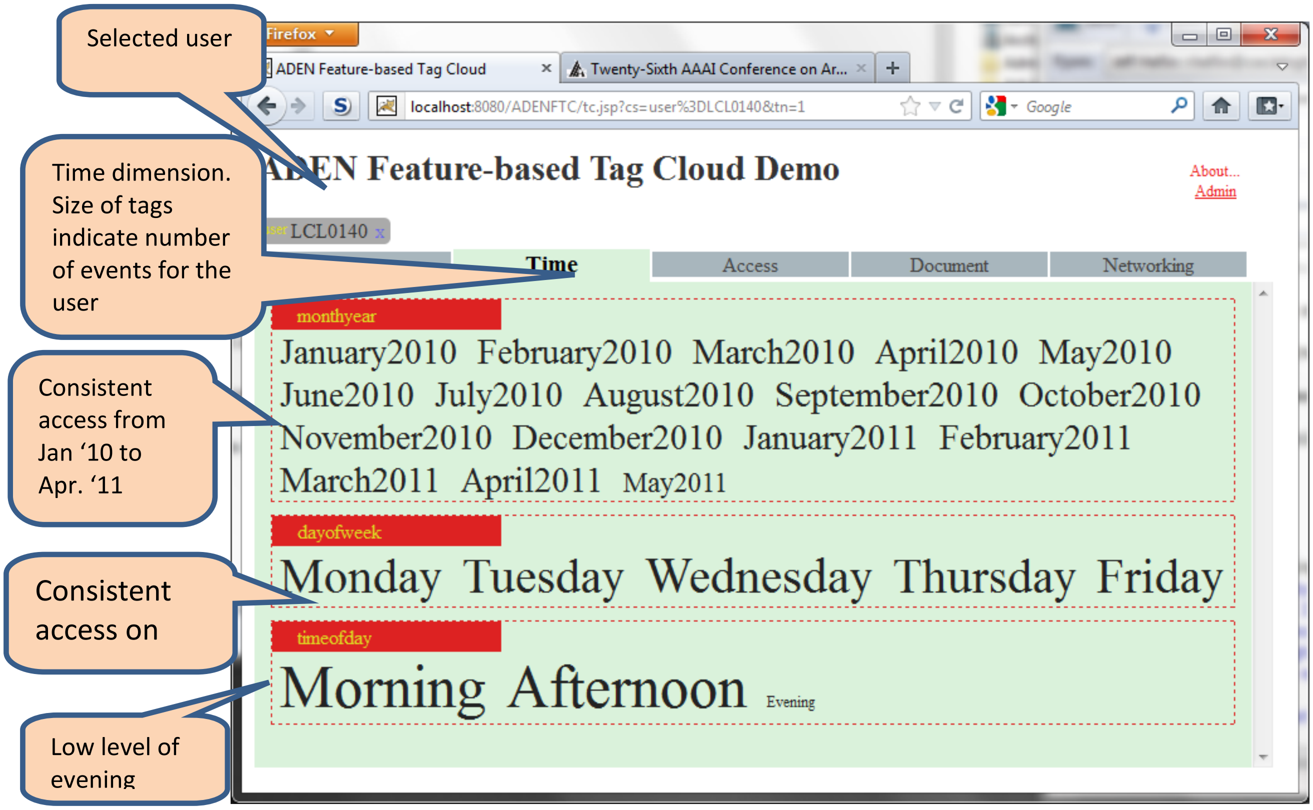
Thus, we are in the process of designing a *Combined Codebook* and either developing or leveraging techniques we are building in our lab in order to extract the values of these variables for individual users over time.

So far, our main efforts with regard to the SureView data are related to implementing a new interface (to transform the SureView in our custom codebook data format) and setting of our analysis environment at Raytheon.

ANALYST EXPLANATION ENGINE: TAG CLOUD BROWSER

To complement our detector engine, we developed a Tag Cloud Explanation Engine as a second interface to the user data. This tag cloud visualization provides security officers with an intuitive access to the data. It lets them explore patterns and deviations trough easy-to-use drill-down / roll-up features. This user interface is intended to support improved visual analytics, i.e. it should assist humans in making sense of the pro-processed data.

We spend much effort on speed and usability to create a good user experience. The high false positive rate of anomaly detection needs to be complemented with a tool that help security analyst to investigate the alerts.


Firefox
ADEN Feature-based Tag Cloud
Twenty-Sixth AAAI Conference on Ar...
localhost:8080/ADENFTC/tc.jsp?cs=user%3DLCL0140&tn=1
Google
ADEN Feature-based Tag Cloud Demo
About...
Admin
user LCL0140 x
Time
Access
Document
Networking
monthyear
January2010 February2010 March2010 April2010 May2010 June2010 July2010 August2010 September2010 October2010 November2010 December2010 January2011 February2011 March2011 April2011 May2011
dayofweek
Monday Tuesday Wednesday Thursday Friday
timeofday
Morning Afternoon Evening

Selected user

Time dimension. Size of tags indicate number of events for the user

Consistent access from Jan '10 to Apr. '11

Consistent access on

Low level of evening

Figure 2. Screenshot of System

II. REFERENCES

1. M. Ovelgionne, C. Kang, A. Sawant and V.S. Subrahmanian. Covertness Centrality in Networks, Proc. 2012 Intl. Symposium on Foundations of Open Source Intelligence and Security Informatics (FOSINT-SI), Istanbul, Turkey, August 2012.

2. M. Broecheler, A. Pugliese, and V.S. Subrahmanian. Efficient Multi-View Maintenance in the Social Semantic Web, WWW (Companion Volume) 2012.

3. C. Kang, J. Grant, A. Pugliese, and V.S. Subrahmanian. STUN: Spatio-Temporal Uncertain (Social) Networks, accepted for publication at 2012 International Conference on Advances in Social Network Analysis and Mining (ASONAM 2012), August 2012, Istanbul, Turkey (full paper – 16% acceptance rate).

4. P. Shakarian, M. Broecheler, V.S. Subrahmanian and C. Molinaro. Social Network Diffusion Optimization Problems, accepted for publication in *ACM Transactions on Computational Logic*, summer 2012.